Real-time Hand Tracking Using a Sum of Anisotropic Gaussians Model

Srinath Sridhar[1], Helge Rhodin[1], Hans-Peter Seidel[1], Antti Oulasvirta[2], Christian Theobalt[1]

[1]Max Planck Institute for Informatics
Saarbrücken, Germany
{ssridhar,hrhodin,hpseidel,theobalt}@mpi-inf.mpg.de

[2]Aalto University
Helsinki, Finland
antti.oulasvirta@aalto.fi

Abstract

Real-time marker-less hand tracking is of increasing importance in human-computer interaction. Robust and accurate tracking of arbitrary hand motion is a challenging problem due to the many degrees of freedom, frequent self-occlusions, fast motions, and uniform skin color. In this paper, we propose a new approach that tracks the full skeleton motion of the hand from multiple RGB cameras in real-time. The main contributions include a new generative tracking method which employs an implicit hand shape representation based on Sum of Anisotropic Gaussians *(SAG), and a pose fitting energy that is smooth and analytically differentiable making fast gradient based pose optimization possible. This shape representation, together with a full perspective projection model, enables more accurate hand modeling than a related baseline method from literature. Our method achieves better accuracy than previous methods and runs at 25 fps. We show these improvements both qualitatively and quantitatively on publicly available datasets.*

1. Introduction

Marker-less articulated hand motion tracking has important applications in human–computer interaction (HCI). Tracking all the degrees of freedom (DOF) of the hand for such applications is hard because of frequent self-occlusions, fast motions, limited field-of-view, uniform skin color, and noisy data. In addition, these applications impose constraints on tracking, including the need for *real-time* performance, high *accuracy*, *robustness*, and low *latency*. Most approaches from the literature thus frequently fail on even moderately fast and complex hand motion.

Previous methods for hand tracking can be broadly classified into either *generative* methods [12, 20, 11] or *discriminative* methods [1, 25, 24, 8]. Generative methods usually employ a dedicated model of hand shape and articulation whose pose parameters are optimized to fit image data. While this yields temporally smooth solutions, real-time performance necessitates fast local optimization strategies which may converge to erroneous local pose optima. In contrast, discriminative methods detect hand pose from image features, *e.g.*, by retrieving a plausible hand configuration from a learned space of poses, but the results are usually temporally less stable.

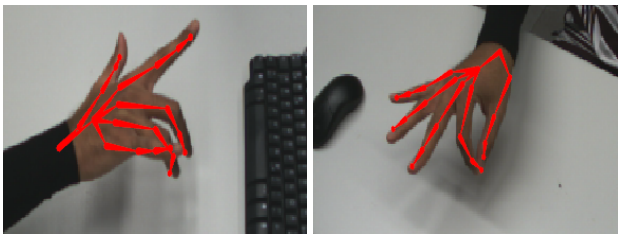

Figure 1. Qualitative tracking results from our SAG-based tracking method. We achieve a framerate of 25 fps which is suitable for interaction applications.

Recently, a promising hybrid method has been proposed that combines generative and discriminative pose estimation for hand tracking from multi-view video and a single depth camera [19]. In their work, generative tracking is based on an implicit Sum of Gaussians (SoG) representation of the hand, and discriminative tracking uses a linear SVM classifier to detect fingertip locations. This approach showed increased tracking robustness compared to prior work but was limited to using isotropic Gaussian primitives to model the hand.

In this paper, we build on this previous method and further develop it to enable fast, more accurate, and robust articulated hand tracking at real-time rates of 25 fps. We contribute a fundamentally extended generative tracking algorithm based on an augmented implicit shape representation.

The original SoG model is based on the simplifying assumption that all Gaussians in 3D have **isotropic covariance**, facilitating simpler projection and energy computation. However, in the case of hand tracking this isotropic 3D SoG model reveals several disadvantages. Therefore we introduce a new 3D *Sum of Anisotropc Gaussians* (SAG)

representation (Figure 3) that uses **anisotropic** 3D Gaussian primitives attached to a kinematic skeleton to approximate the volumetric extent and motion of the hand. This step towards a more general class of 3D functions complicates the projection from 3D to 2D and thus the computation of the pose fitting energy. However, it maintains important smoothness properties and enables a better approximation of the hand shape with less primitives (visualized as ellipsoids in Figure 3). Our approach, in contrast to previous methods [21, 19], models the full perspective projection of 3D Gaussians. To summarize, the primary contributions of our paper are:

- An advancement of [19] that generalizes the SoG-based tracking to one based on a new 3D Sum of Anisotropic Gaussians (SAG) model, thus enabling tracking using fewer primitives.
- Utilization of a full perspective projection model for projection of 3D Gaussians to 2D in matrix-vector form.
- Analytic derivation of the gradient of our pose fitting energy, which is smooth and differentiable, to enable real-time optimization.

We evaluate the improvements enabled by SAG-based generative tracking over previous work. Our contributions not only lead to more accurate and robust real-time tracking but also allow tracking of objects in addition to the hand.

2. Previous Work

Following the survey of Erol *et al*. [5] we review previous work by categorizing them into either *model-based tracking* methods or *single frame pose estimation* methods. Model-based tracking methods use a hand model, usually a kinematic skeleton with additional surface modeling, to estimate the parameters that best explain temporal image observations. Single frame methods are more diverse in their algorithmic recipes, they make fewer assumptions about temporal coherence and often use non-parametric models of the hand. Hand poses are inferred by exploiting some form of inverse mapping from image features to a space of hand configurations.

Model-based Tracking: Rehg and Kanade [15] were one of the first to present a kinematic model-based hand tracking method. Lin *et al*. [10, 28] studied the constraints of hand motion and proposed feasible base states to reduce the search space size. Oikonomidis *et al*. [12] presented a method based on particle swarm optimization for full DoF hand tracking using a depth sensor and achieved a frame rate of 15 fps with GPU acceleration. Other model-based methods using global optimization for pose inference fail to perform at real-time frame rates [20, 11].

Primitive shapes such as spheres and (super-)quadrics have been explored for tracking objects [9], and, recently, for tracking hands [14]. However, perspective projection of complex shapes is hard to represent analytically and therefore fast optimization is hard. In this work we use anisotropic Gaussian primitives with analytical expression for perspective projection. An overview of perspective projection of spheroids, which are conceptually similar to anisotropic Gaussians, can be found in [4].

Tracking hands with objects imposes additional constraints on hand motion. Methods proposed by Hamer *et al*. [7, 6], and others [13, 16, 3] model these constraints. However, these methods require offline computation and are unsuitable for interaction applications.

Single Frame Pose Estimation: Single frame methods estimate hand pose in each frame of the input sequence without taking temporal information into account. Some methods build an exemplar pose database and formulate pose estimation as a database indexing problem [1]. The retrieval of the whole hand pose was explored by Wang and Popović [25, 24]. However, the hand pose space is large and it is difficult to sample it with sufficient granularity for jitter-free pose estimation. Sridhar *et al*. [19] proposed a part-based pose retrieval method to reduce the search space. Decision and regression forests have been successfully used in full body pose estimation to learn human pose from a large synthetic dataset [18]. This approach has been recently adopted for hands [8, 23, 29, 22]. These methods generally lack temporal stability and recover only joint positions or part labels instead of a full kinematic skeleton.

Hybrid Tracking: Hybrid frameworks that combine the advantages of model-based tracking and single frame pose estimation can be found in full body pose estimation [30, 2, 26] and early hand tracking [17]. A hybrid method that uses color and depth data for hand tracking was proposed [19]. However, this method is limited to studio conditions and uses isotropic Gaussian primitives. In this paper, we extend their method by introducing an improved (model-based) generative tracker. This new tracker alone leads to higher tracking accuracy and robustness than the baseline method it extends.

3. Tracking Overview

Figure 2 shows an overview of our tracking framework. The goal is to estimate hand pose robustly by maximizing the similarity between the hand model and the input images. The tracker developed in this paper extends the generative pose estimation method of the tracking algorithm in [19].

The input to our method are a set of RGB images from 5 video cameras (Point Grey Flea 3) of resolution 320×240 (see Figure 2). The cameras are calibrated and run at 60 fps. The hand is modeled as a full kinematic skeleton with 26 degrees-of-freedom (DOF), and unlike other methods that deliver only joint locations or part labels in the images, our approach computes full kinematic joint angles $\Theta^* = \{\theta_j^*\}$.

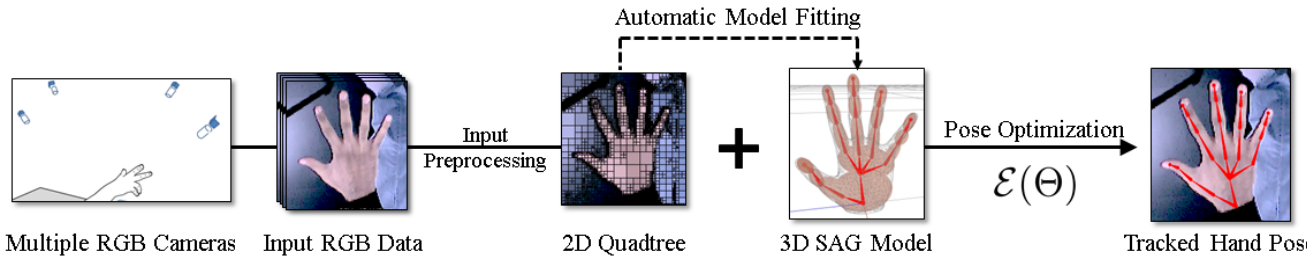

Figure 2. Overview of the our tracking framework. We present a novel generative tracking method that models the hand as a *Sum of Anisotropic Gaussians*. We obtain better accuracy and robustness than previous work.

Our method maximizes the similarity between the 3D hand model projected into all RGB images, and the images themselves, by means of a fast iterative local optimization algorithm. The main novelty and contribution of this work is the new shape representation and pose optimization framework used in the tracker (Section 4). Our pose fitting energy is smooth, differentiable and allows fast gradient-based optimization. This enables real-time hand tracking with higher accuracy and robustness while using fewer Gaussian primitives.

4. SAG-based Generative Tracking

The generative tracker from [19] is based on a representation called a Sum of Gaussians (SoG) model, that was originally proposed in [21] for full body tracking. The basic concept of the SoG model is to approximate the 3D volumetric extent of the hand by **isotropic Gaussians** attached to the bones of the skeleton, with a color associated to each Gaussian (Figure 3 (c)). Similarly, input images are segmented to regions of coherent color, and each region is approximated by a 2D SoG (Figure 4 (b-c)). A SoG-based pose fitting energy was defined by measuring the overlap (in terms of spatial support and color) between the projected 3D Gaussians and all the image Gaussians. This energy is maximized with respect to the degrees of freedom to find the correct pose.

Unfortunately, a faithful approximation of the hand volume with a collection of isotropic 3D Gaussians often requires many Gaussian primitives with small standard deviation, a problem akin to packing a volume with spherical primitives. With SoG, this leads to sub-optimal hand shape approximation and increased computational complexity due to a high number of primitives in 3D (Figure 3). In this paper, we extend the SoG model and represent the hand shape in 3D with **anisotropic Gaussians**, yielding a *Sum of Anisotropic Gaussians* model (see Figure 1). This not only enables a better approximation of the hand shape with less 3D primitives (Figure 3), but also leads to higher pose estimation accuracy and robustness. The move to anisotropic 3D Gaussians complicates their projection into 2D where scaled orthographic projection [21, 19] cannot be used. But we show that the numerical benefits of the SoG representation hold equally for the SAG model: 1) We derive a pose fitting energy that is smooth and analytically differentiable for the SAG model under perspective projection that allows efficient optimization with a gradient-based iterative solver; 2) We show that occlusions can be efficiently approximated with the SAG model within our energy formulation. This is in contrast to many other generative trackers where occlusion handling leads to discontinuous pose fitting energies.

4.1. Fundamentals of SAG Model

We represent both the volume of the hand in 3D, as well as the RGB images with a collection of anisotropic Gaussian functions. A *Sum of Anisotropic Gaussians* (SAG) model thus takes the form:

$$\mathcal{C}(\mathbf{x}) = \sum_{i=1}^{n} \mathcal{G}_i(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{1}$$

where $\mathcal{G}_i(.)$ denotes a un-normalized, anisotropic Gaussian

$$\mathcal{G}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) := \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T {\boldsymbol{\Sigma}_i}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right], \tag{2}$$

with mean $\boldsymbol{\mu}_i$ and covariance matrix is $\boldsymbol{\Sigma}_i$ for the i^{th} Gaussian. Each Gaussian also has an associated average color vector $\mathbf{c}_i$ in HSV color space.

Using the above representation, we model the hand surface as a sum of 3D anisotropic Gaussians (3D SAG), where $\mathbf{x} \in \mathbb{R}^3$. We also approximate the input RGB images as a sum of 2D isotropic Gaussians (2D SoG), where $\mathbf{x} \in \mathbb{R}^2$. This is an extension of the SoG representation proposed earlier in [21, 19], which was limited to isotropic Gaussians.

3D Hand Modeling: We model the volumetric extent of the hand as a 3D sum of anisotropic Gaussians model (3D SAG), where $\mathbf{x} \in \mathbb{R}^3$. Each $\mathcal{G}_i$ in the 3D SAG is attached to one bone of the skeleton, and thus moves with the local frame of the bone (Figure 3). A linear mapping between skeleton joint angles and a pose parameter space, $\Theta = \{\theta_j\}$, is constructed. The skeleton pose parameters are further

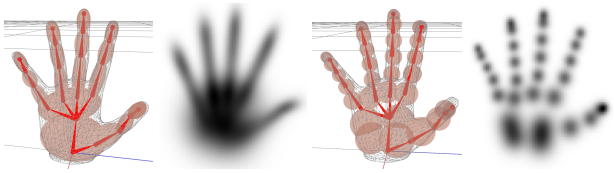

Figure 3. (a) Our 3D SAG hand model with 17 **anisotropic Gaussians** visualized as ellipsoids with radii equal to the standard deviation. (c) A 3D SoG hand model with 30 **isotropic Gaussians** visualized as spheres. (b) Visualizes the SAG model density when projected into 2D; with less primitives, the shape of the hand is much better approximated than with the SoG model (d) [19].

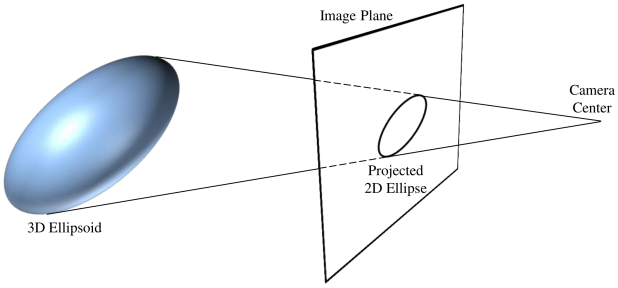

Figure 4. Sketch of the perspective projection of an ellipsoids as the intersection of the image plane with the cone formed by the camera center and the ellipsoid.

constrained to a predefined range of motion reflecting human anatomy, $\theta_j \in [l^j_{min}, l^j_{max}]$. The use of anisotropic Gaussians, whose spatial density is controlled by the covariances, enables us to approximate the general shape of the hand with less primitives than needed with the original isotropic SoG model (Figure 3). This is because we can create a better *packing* of the hand volume with more generally elongated Gaussians, particularly for approximating cylindrical structures like the fingers. The Gaussians in 3D have infinite spatial support, which is an advantageous property for pose fitting, as explained later, but also means that the SAG does not represent a finite volume ($\mathcal{C}(\mathbf{x}) > 0$ everywhere). We therefore assume that the hand is well modeled by a 3D SAG if the surface passes through each Gaussian at a distance of 1 standard deviation from the mean.

Hand Model Initialization: The hand model for tracking requires initialization of the skeleton dimensions, Gaussian covariances that control their shapes, and associated colors for an actor before it can be used for tracking. Our method accepts manually created hand models which could be obtained from a laser scan. Alternatively, we also provide a fully automatic procedure to obtain a hand model to fit each person. This method uses a greedy optimization strategy to optimize for a total of 3 global hand shape parameters and 3 independent scaling parameters (along the local principal axes) for each of the 17 Gaussians in the hand model. We observed that starting with a manual model and then using the greedy fitting algorithm works best.

2D RGB Image Modeling: We approximate the input RGB images using 2D SoG, $\mathcal{C}_I$, by quad-tree clustering of regions of similar color. While it would also be possible to approximate the image as 2D SAG, the computational expense of the non-uniform region segmentation would prohibit realtime performance. We found in our experiments that around 500 2D image Gaussians were generated for each camera image.

4.2. Projection of 3D SAG to 2D SAG

Pose optimization (Section 4.3) requires the comparison of the projections of the 3D SAG into all camera views, with the 2D SoG of each RGB image. Intuitively (for a moment ignoring infinite support), SAG and SoG can be visualized as ellipsoids and spheres, respectively.

The perspective projections of spheres and ellipsoids both yield ellipses in 2D [4]. For the case of isotropic Gaussians in 3D, like in the earlier SoG model, projection of a 3D Gaussian can be approximated as a 2D Gaussian with a standard deviation that is a scaled orthographic projection of the 3D standard deviation [19, 21]. This simple approximation does not hold for our anisotropic Gaussians.

Therefore, we utilize an exact perspective projection Π of ellipsoids, in order to model the projection of the 3D SAG hand model, $\mathcal{C}_H$, into its 2D SAG equivalent, $\mathcal{C}_P$. Given an ellipsoid in $\mathbb{R}^3$ with associated mean, $\boldsymbol{\mu}_h$ and covariance matrix, $\boldsymbol{\Sigma}_h$, its perspective projection can be visualized as an ellipse in $\mathbb{R}^2$ with parameters $\boldsymbol{\Sigma}_p$ and mean $\boldsymbol{\mu}_p$. Figure 4 (a) sketches the perspective projection, intuitively, Π can be thought of as the intersection of the elliptical cone (formed by the ellipsoid and the camera center) with the image plane. Without loss of generality, we assume the camera to be at the origin with a camera matrix, $\mathbf{P} = \mathbf{K}\,[\,\mathbf{I}\,|\,\mathbf{0}\,]$. The parameters of the projected Gaussian are given by

$$\boldsymbol{\mu}_p = \frac{1}{|\mathbf{M}_{33}|}\,\mathbf{K}_{33}\begin{bmatrix} |\mathbf{M}_{31}| \\ -|\mathbf{M}_{23}| \end{bmatrix} + \begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix}, \tag{3}$$

$$\boldsymbol{\Sigma}_p = -\frac{|\mathbf{M}|}{|\mathbf{M}_{33}|}\,\mathbf{K}_{33}\,\mathbf{M}_{33}^{-1}\,\mathbf{K}_{33}^{T}, \tag{4}$$

where

$$\mathbf{M} = \boldsymbol{\Sigma}_h^{-1}\,\boldsymbol{\mu}_h\boldsymbol{\mu}_h^T\boldsymbol{\Sigma}_h^{-\top} - \left(\boldsymbol{\mu}_h^\top\boldsymbol{\Sigma}_h^{-1}\boldsymbol{\mu}_h - 1\right)\boldsymbol{\Sigma}_h^{-1}, \tag{5}$$

$|\mathbf{M}|$ is the determinant of $\mathbf{M}$, $\mathbf{A}_{ij}$ is a matrix $\mathbf{A}$ with its i^{th} row and j^{th} column removed, and k_{ij} is the element at the i^{th} row and j^{th} column of $\mathbf{K}$. Please see the supplementary material for the derivation of $\mathbf{M}$ and the projection with arbitrary camera matrices. This more general projection also leads to a more involved pose fitting energy with more involved derivatives than for the SoG model, as explained in the next section.

4.3. Pose Fitting Energy

We now define an energy that measures the quality of overlap between the projected 3D SAG $\mathcal{C}_P = \Pi(\mathcal{C}_H)$, and the image SoG $\mathcal{C}_I$, and that is optimized with respect to the pose parameters Θ of the hand model. Our overlap measure is an extension of the SoG overlap measure [21] to a SAG. Intuitively, we assume that two Gaussians in 2D match well if their spatial support aligns, and their color matches. This criterion can be expressed by the spatial integral over their product, weighted by a color similarity term. The similarity of any two sets, $\mathcal{C}_a$ and $\mathcal{C}_b$, of SAG or SoG in 2D (including combined models with both isotropic and anisotropic Gaussians) can thus be defined as

$$\begin{aligned} E(\mathcal{C}_a, \mathcal{C}_b) &= \sum_{p \in \mathcal{C}_a} \sum_{q \in \mathcal{C}_b} d(\mathbf{c}_p, \mathbf{c}_q) \int_\Omega \mathcal{G}_p(\mathbf{x}) \mathcal{G}_q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \sum_{p \in \mathcal{C}_a} \sum_{q \in \mathcal{C}_b} d(\mathbf{c}_p, \mathbf{c}_q) \, D_{pq} = \sum_{p \in \mathcal{C}_a} \sum_{q \in \mathcal{C}_b} E_{pq} \end{aligned} \quad (6)$$

where E_{pq} is the integral overlap measure mentioned earlier, $d(\mathbf{c}_p, \mathbf{c}_q)$ measures color similarity using the Wendland function [27], and $E_{pq} = d(\mathbf{c}_p, \mathbf{c}_q) \, D_{pq}$. Unlike the SoG model, for the general case of potentially anisotropic Gaussians, the term D_{pq} evaluates to

$$D_{pq} = \frac{\sqrt{(2\pi)^2 |\boldsymbol{\Sigma}_p \, \boldsymbol{\Sigma}_q|}}{\sqrt{|(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q)|}} \mathbf{e}^{-\frac{1}{2}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^T (\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q)^{-1} (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)}. \quad (7)$$

Using this Gaussian similarity formulation allows us to compute the similarity between the image SoG $\mathcal{C}_I$ and the projected hand SAG $\mathcal{C}_P$.

We also need to consider occlusions of Gaussians from a camera view. Computing a function that indicates occlusion analytically independent of pose parameters is generally difficult and may lead to a discontinuous similarity function. Thus, we use a heuristic approximation of occlusion [21] that yields a continuous fitting energy defined as follows

$$E_{sim}\left[\mathcal{C}_I, \mathcal{C}_H\right] = \sum_{q \in \mathcal{C}_I} \min \left(\sum_{p \in \Pi(\mathcal{C}_H)} w_p^h \, E_{pq}, E_{qq} \right), \quad (8)$$

where w_p^h is a weighting factor for each projected 3D Gaussian of the hand model. E_{qq} is the overlap of an image Gaussian with itself. With this formulation, an image Gaussian cannot contribute more to the overlap similarity than by its own footprint in the image. To find the hand pose, E_{sim} is optimized with respect to Θ, as described in the following section. Note that the infinite support of the Gaussians is advantageous as it leads to an *attracting force* between the projected model and the image of the hand, even if they do not overlap in a camera view.

4.4. Pose Optimization

The final energy that we maximize to find the hand pose takes the form

$$\mathcal{E}(\Theta) = E_{sim}(\Theta) - w_l \, E_{lim}(\Theta), \quad (9)$$

where $E_{lim}(\Theta)$ penalizes motions outside of parameter limits quadratically, and weight w_l is empirically set to 0.1. With the SoG formulation, it was possible to express the energy function (with a scaled orthographic projection) in a closed form analytic expression, and to derive the analytic gradient. We have found that $E_{sim}(\Theta)$ in our SAG-based, even with its full perspective projection model, can still be written in closed form with analytic gradient.

We derive the analytical gradient of E_{sim} with respect to the degrees of freedom Θ in three steps. For each Gaussian pair (h, q) and parameter θ_j we compute

$$\left(\frac{\partial \boldsymbol{\Sigma}_h}{\partial \theta_j}, \frac{\partial \boldsymbol{\mu}_h}{\partial \theta_j} \right) \xrightarrow{a)} \left(\frac{d\boldsymbol{M}}{d\theta_j} \right) \xrightarrow{b)} \left(\frac{d\boldsymbol{\Sigma}_p}{d\theta_j}, \frac{d\boldsymbol{\mu}_p}{d\theta_j} \right) \xrightarrow{c)} \left(\frac{dD_{pq}}{d\theta_j} \right). \quad (10)$$

We exemplify the computation at hand of step a); the input to a) is the change of the ellipsoid covariance matrix $\partial \boldsymbol{\Sigma}_h^{-1}$ and the change of position $\partial \boldsymbol{\mu}_h$ with respect to the DOF θ_j. In this step we are interested in the total derivative

$$\begin{aligned} \frac{d\boldsymbol{M}}{d\theta_j} = &\sum_{i \in \{1,2,3\}} \frac{\partial \mathbf{M}}{\partial \boldsymbol{\mu}_{h_i}} \frac{\partial \boldsymbol{\mu}_{h_i}}{\partial \theta_j} \\ &+ \sum_{k \in \{1, \cdots, 6\}} \frac{\partial \mathbf{M}}{\partial \boldsymbol{\Sigma}_{h_k}^{-1}} \frac{\partial \boldsymbol{\Sigma}_{h_k}^{-1}}{\partial \theta_j}. \end{aligned} \quad (11)$$

Following matrix calculus, the partial derivatives of the cone matrix $\boldsymbol{M}$ with respect to $\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h$ are

$$\begin{aligned} \frac{\partial \mathbf{M}}{\partial \boldsymbol{\mu}_{h_i}} &= \boldsymbol{\Sigma}_h^{-1} (\mathbf{e}^i \boldsymbol{\mu}_h^\top + \mathbf{e}^i \boldsymbol{\mu}_h^\top)^\top \boldsymbol{\Sigma}_h^{-1} \\ &\quad - \left((\mathbf{e}^{i\top} \boldsymbol{\Sigma}_h^{-1} \boldsymbol{\mu}_h) + (\mathbf{e}^{i\top} \boldsymbol{\Sigma}_h^{-1} \boldsymbol{\mu}_h)^\top \right) \boldsymbol{\Sigma}_h^{-1}, \\ \frac{\partial \mathbf{M}}{\partial \boldsymbol{\Sigma}_{h_k}^{-1}} &= H + H^\top - \boldsymbol{\mu}_h{}^\top \mathbf{S}^k \boldsymbol{\mu}_h \boldsymbol{\Sigma}_h^{-1} \\ &\quad - (\boldsymbol{\mu}_h^\top \boldsymbol{\Sigma}_h^{-1} \boldsymbol{\mu}_h - 1) \mathbf{S}^k, \end{aligned} \quad (12)$$

with e^i the i^{th} unit vector, $\boldsymbol{\mu}_{h_i}$ the i^{th} entry of $\boldsymbol{\mu}_h$, $\boldsymbol{H} = \mathbf{S}^k \boldsymbol{\mu}_h \boldsymbol{\mu}_h^\top \boldsymbol{\Sigma}_h^{-\top}$, where k indexes the unique elements of the symmetric matrix $\boldsymbol{\Sigma}_h^{-1}$, and $\mathbf{S}^k$ is the symmetric structure matrix with the k^{th} elements equal to one, and zero otherwise. Steps b) and c) are derived in a similar manner and can be found in the supplementary document. The total similarity energy E_{sim} is the weighted sum over all θ_j and D_{pq} according to equation 8. Combined with the analytical gradient of $E_{lim}(\Theta)$ we obtain an analytic formulation for $\frac{\partial}{\partial \Theta} \mathcal{E}(\Theta)$. As sums of independent terms, both $\mathcal{E}$ and $\frac{\partial}{\partial \Theta} \mathcal{E}$ lend themselves to parallel implementation.

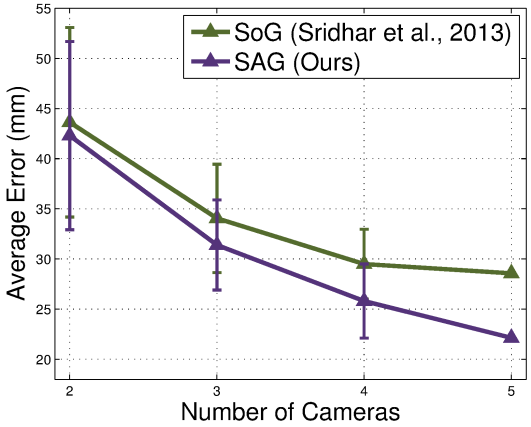

Figure 5. This figure shows a comparison of tracking error for SAG and SoG with 2 to 5 cameras. A total of 156 runs were required for SAG and SoG with different camera combinations. The results show that SAG outperforms SoG. Best viewed in color.

Even though evaluation of fitting energy and gradient is much more involved than for the SoG model, both share the same smoothness properties and can be evaluated efficiently, and thus an optimal pose estimate can be computed effectively using a standard gradient-based optimizer. The optimizer is initiliazed with an extrapolation of the pose parameters from the two previous time steps. The SAG framework leads to much better accuracy and robustness and requires far fewer shape primitives to be compared, as validated in Section 5.

5. Experiments

We conducted extensive experiments to show that our SAG-based tracker outperforms the SoG based method it extends. We also compare with another state-of-the-art method that uses a single depth camera [22]. We ran our experiments on the publicly available **Dexter 1** dataset [19] which has ground truth annotations. This dataset contains challenging, slow and fast motions. We processed all 7 sequences in the dataset and, while [19] evaluated their algorithm only on the slow motions, we evaluated our method on fast motions as well.

For all results we used 10 gradient ascent iterations. Our method runs at a framerate of 25 fps on an Intel Xeon E5-1620 running at 3.60 GHz with 16 GB RAM. Our implementation of the SoG-based tracker of [19] runs slightly faster at 40 fps.

Accuracy: Figure 6 shows a plot of the average error for each sequence in our dataset. Over all sequences, SAG had an error of 24.1 mm, SoG had an error of 31.8 mm, and [22] had an error of 42.4 mm (only 3 sequences). The mean standard deviations were 11.2 mm for SAG, 13.9 mm for SoG, and 8.9 mm for [22] (3 sequences only). Our errors are higher than those reported by [19] because we performed our experiments on both the slow and fast motions as opposed to slow motions only. Additionally, we discarded the palm center used by [19] since this is not clearly defined. We would like to note that [22] perform their tracking on the depth data in Dexter 1 , and use no temporal information. In summary, SAG achieves the lowest error and is 7.7 mm better than SoG. This improvement is nearly the width of a finger thus making it a significant gain in accuracy.

Error Frequency: Table 1 shows an alternative view of the accuracy and robustness improvement of SAG. We calculated the percentage of frames of each sequence in which the tracking error is less than x mm where $x \in \{15, 20, 25, 30, 45\}$. This experiment shows clearly that SAG outperforms SoG in almost all sequences and error bounds. In particular the improvement in **accuracy** is measured by the increased number of frames with error smaller than 15 mm, and the **robustness to fast motions** by the smaller number of dramatic failures of errors larger than 30 mm. For example, in the adbadd sequence 70.7% of frames are better than 15 mm for SAG while only 34.5% of frames for SoG. Note that when $x = 100$ mm, the percentage of frames $< x$ mm is 100% for SAG.

Effect of Number of Cameras: To evaluate the scalability of our method to the number of cameras we conducted an experiment where each camera was progressively disabled with total active cameras ranging from 2 to 5. This leads to 26 possible camera combinations for each sequence and a total of 156 runs for both the SAG and SoG methods. We excluded the random sequence as it was too challenging for tracking with 3 or less cameras.

Figure 5 shows the average error over all runs for varying cameras. Clearly, SAG produces lower errors and standard deviations for all camera combinations. We also observe a diverging trend and hypothesize that as the number of cameras is increased the gap between SAG and SoG will also increase. This may be important for applications requiring very precise tracking such as motion capture for movies. We associate the improvements in accuracy of SAG with its ability to approximate the users' hand better than SoG. Figure 3 (b, d) visualizes the projected model density and reveals a better approximation for SAG.

Qualitative Tracking Results: Finally, we show several qualitative results of tracking in Figure 7 comparing SAG and SoG. Since our tracking approach is flexible we are also able to track additional simple objects such as a plate using only a few primitives. These additional results, qualitative comparison to [22], and failure cases can be found in the supplementary video.

6. Discussion and Future Work

As demonstrated in the above experiments our method advances state of the art methods in accuracy and is suitable for real-time applications. However, the generative method

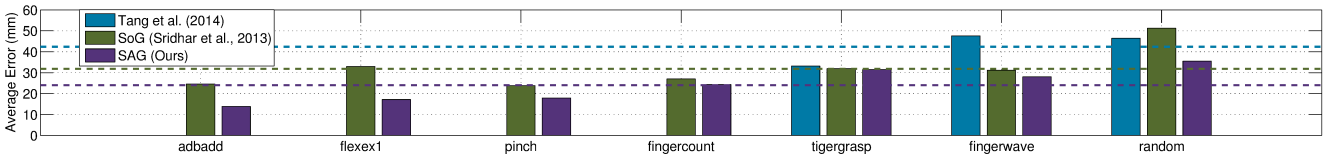

Figure 6. Average errors for all sequences in the Dexter 1 dataset. Our method has the lowest average error of 24.1 mm compared to SoG (31.8 mm) and [22] (42.4 mm). The dashed lines represent average errors over all sequences. Best viewed in color.

Error $< x$ mm	**adbadd**		**fingercount**		**fingerwave**		**flexex1**		**pinch**		**random**		**tigergrasp**	
	SoG	SAG	SoG	SAG	SoG	SAG	SoG	SAG	SoG	SAG	SoG	SAG	SoG	SAG
15	34.5	**70.7**	**13.1**	8.7	11.0	**16.7**	5.2	**50.0**	10.8	**34.0**	3.3	**10.5**	**11.2**	10.2
20	48.1	**97.5**	**35.2**	33.4	31.0	**34.3**	12.1	**79.4**	30.78	**66.3**	5.3	**21.4**	**25.6**	**25.6**
25	61.0	**99.4**	54.1	**61.0**	45.8	**47.0**	29.7	**91.0**	50.3	**89.9**	6.9	**34.7**	43.8	**51.7**
30	70.7	**99.4**	65.4	**79.4**	58.5	**59.4**	45.0	**96.5**	81.0	**98.7**	10.9	**46.4**	50.2	**58.7**
45	93.4	**99.7**	90.1	**99.1**	82.0	**90.4**	86.6	**98.9**	**100.0**	**100.0**	40.2	**72.1**	**83.3**	82.3

Table 1. Percentage of total frames in a sequence that have an error of less x mm. We observe that SAG outperforms SoG in all sequences and error bounds. The values in bold face indicate the best values for a given error bound.

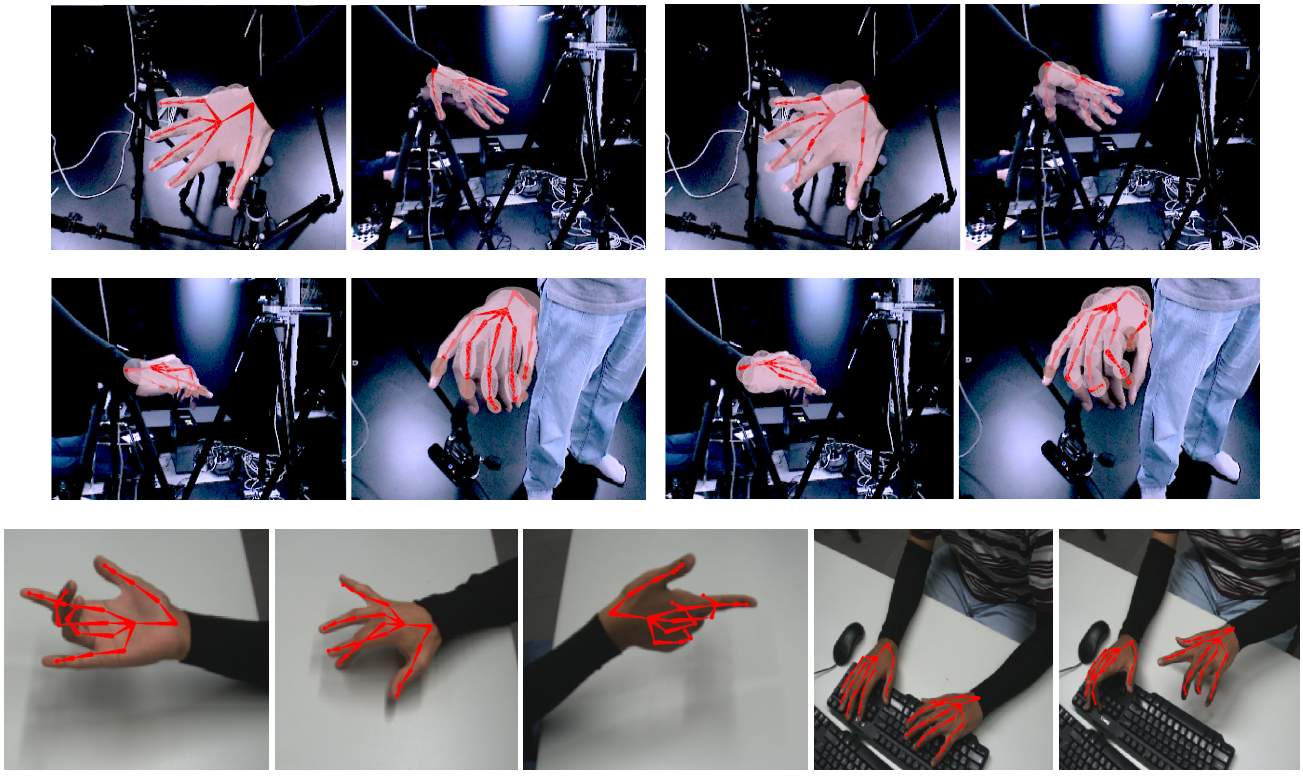

Figure 7. **First Two Rows**: Comparison of SAG (left) and SoG (right) for two frames in the Dexter 1 dataset. In the first row, SAG covers the hand much better during a fast motion of the hand in spite of using fewer primitives. In the second row, a challenging motion is performed for which SAG performs better. **Bottom Row**: Realtime tracking results for one hand with different actors, and two hands. Please see supplementary video for results from hand + object tracking.

can lose tracking because of fast hand motions. Like other hybrid methods, we could augment our method with a discriminative tracking strategy. The generality of our method allows easy integration into such a hybrid framework.

In terms of utility, we require the user to wear a black sleeve and we use multiple calibrated cameras. These limitations could be overcome if we would only rely the depth data for our tracking. Since the SAG representation is data agnostic, we could model the depth image as a SAG as well. We intend to explore these improvements in the future.

7. Conclusion

We presented a method for articulated hand tracking that uses a novel Sum of Anisotropic Gaussians (SAG) representation to track hand motion. Our SAG formulation uses a full perspective projection model and uses only a few Gaussians to model the hand. Because of our smooth and differentiable pose fitting energy, we are able to perform fast gradient-based pose optimization to achieve real-time frame rates. Our approach produces more robust and accurate tracking than previous methods while featuring advantageous numerical properties and comparable runtime. We demonstrated our accuracy and robustness on standard datasets by comparing with relevant work from literature.

Acknowledgments: This work was supported by the ERC Starting Grant CapReal. We would like to thank James Tompkin and Danhang Tang.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proc. of IEEE CVPR 2003*, pages 432–9 vol.2. 1, 2

[2] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. of ICCV 2011*, pages 1092 –1099. 2

[3] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *Proc. of ECCV 2012*, pages 640–653 v.7577. 2

[4] D. Eberly. Perspective projection of an ellipsoid. http://www.geometrictools.com/, 1999. 2, 4

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, pages 52–73 v.108, 2007. 2

[6] H. Hamer, J. Gall, T. Weise, and L. Van Gool. An object-dependent hand pose prior from sparse training data. In *Proc. of CVPR 2010*, pages 671–678. 2

[7] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *Proc. of ICCV 2009*, pages 1475–1482. 2

[8] C. Keskin, F. Kra, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proc. of ECCV 2012*, pages 852–863. 1, 2

[9] J. Krivic and F. Solina. Contour based superquadric tracking. Lecture Notes in Computer Science, pages 1180–1186. 2003. 2

[10] J. Lin, Y. Wu, and T. Huang. Modeling the constraints of human hand motion. In *Workshop on Human Motion, 2000. Proceedings*, pages 121 –126. 2

[11] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. of ECCV 2000*, pages 3–19. 1, 2

[12] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proc. of BMVC 2011*, pages 101.1–101.11. 1, 2

[13] I. Oikonomidis, N. Kyriazis, and A. Argyros. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Proc. of ICCV 2011*, pages 2088–2095. 2

[14] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *Proc. of CVPR 2014*. 2

[15] J. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Proc. of ECCV 1994*, volume 801, pages 35–46 v.801. 2

[16] J. Romero, H. Kjellstrom, and D. Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *Proc. of IEEE ICRA 2010*, pages 458–463. 2

[17] R. Rosales and S. Sclaroff. Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, pages 251–276 v.67, 2006. 2

[18] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. of CVPR 2011*, pages 1297–1304. 2

[19] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *Proc. of ICCV 2013*. 1, 2, 3, 4, 6

[20] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE TPAMI*, 28:1372–1384, 2006. 1, 2

[21] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *Proc. of ICCV 2011*, pages 951 –958. 2, 3, 4, 5

[22] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proc. of CVPR 2014*. 2, 6, 7

[23] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proc. of ICCV 2013*. 2

[24] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proc. of ACM UIST 2011*, pages 549–558. 1, 2

[25] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM TOG*, page 63:163:8 v.28, 2009. 1, 2

[26] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM TOG*, pages 188:1–188:12 v.31, 2012. 2

[27] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math*, 4(1):389–396, 1995. 5

[28] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *Proc. of ICCV 2001*, volume 2, pages 426–432 vol.2. 2

[29] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *Proc. of ICCV 2013*. 2

[30] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *Proc. of ICCV 2011*, pages 731–738. 2